

## Claims

We claim:

1. A computer-implemented method for programmatically generating a graphical program, the method comprising:
  - displaying a graphical user interface (GUI) on a display;
  - receiving user input to the GUI specifying desired functionality of the graphical program;
  - programmatically generating the graphical program in response to the user input specifying the functionality of the graphical program, wherein the graphical program implements the specified functionality.
2. The method of claim 1,
  - wherein the GUI comprises information useable in guiding a user in creation of a program.
3. The method of claim 1,
  - wherein the GUI comprises one or more GUI input panels;
  - wherein the user input to the GUI comprises user input to each of the one or more GUI input panels.
4. The method of claim 3,
  - wherein said displaying the GUI and said receiving user input to the GUI comprise:
    - displaying a first GUI input panel on the display, wherein the first GUI input panel includes one or more first fields adapted to receive user input specifying first functionality of the graphical program;
    - receiving first user input specifying first functionality of the graphical program;

displaying a second GUI input panel on the display, wherein the second GUI input panel includes one or more second fields adapted to receive user input specifying second functionality of the graphical program;

5 receiving second user input specifying second functionality of the graphical program.

5. The method of claim 4,

wherein the second GUI input panel is one of a plurality of possible second GUI input panels, wherein the second GUI input panel is displayed based on the first user  
10 input.

6. The method of claim 1,

wherein said programmatically generating the graphical program comprises programmatically generating a portion of a graphical program.

15 7. The method of claim 1,

wherein said programmatically generating the graphical program creates the graphical program without any user input specifying the new graphical program during said creating.

20 8. The method of claim 1,

wherein the programmatically generated graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program.

25 9. The method of claim 1,

wherein the graphical program comprises a block diagram portion comprising a plurality of interconnected nodes and a user interface portion;

wherein said programmatically generating the graphical program includes generating the block diagram portion and the user interface portion.

10. The method of claim 1, wherein said programmatically generating the graphical program comprises:

5 creating a plurality of graphical program objects in the graphical program; and

interconnecting the plurality of graphical program objects in the graphical program;

wherein the interconnected plurality of graphical program objects comprise at least a portion of the graphical program.

10 11. The method of claim 1, wherein said programmatically generating the graphical program comprises:

15 creating one or more user interface objects in the graphical program, wherein the one or more user interface objects perform one or more of providing input to or displaying output from the graphical program.

12. The method of claim 1,

wherein the user input received specifies an instrumentation function;

wherein the programmatically generated graphical program implements the specified instrumentation function.

20 13. The method of claim 12,

wherein the instrumentation function comprises one or more of:

a test and measurement function; or

an industrial automation function.

25 14. The method of claim 1,

wherein said programmatically generating the graphical program comprises calling an application programming interface (API) enabling the programmatic generation of a graphical program.

15. The method of claim 1,  
wherein said programmatically generating the graphical program comprises  
programmatically requesting a server program to generate the graphical program.

5

16. A computer-implemented method for programmatically generating a  
graphical program, the method comprising:

10 displaying a plurality of GUI input panels on a display, wherein the GUI input  
panels comprise information useable in guiding a user in creation of a program;

receiving user input to the plurality of GUI input panels, wherein the user input  
specifies desired functionality of the graphical program;

15 programmatically generating the graphical program in response to the user input  
specifying the functionality of the graphical program, wherein the graphical program  
implements the specified functionality.

17. A computer-implemented method for programmatically generating a  
20 graphical program, the method comprising:

displaying a graphical user interface (GUI) on a display;

receiving user input to the GUI specifying desired functionality of the graphical  
program;

executing a graphical program generation (GPG) program;

25 the GPG program receiving the user input, wherein the user input specifies the  
desired functionality of the new graphical program;

the GPG program programmatically generating the graphical program in response  
to the user input specifying the functionality of the graphical program, wherein the  
graphical program implements the specified functionality.

18. A computer-implemented method for programmatically generating a graphical program, the method comprising:

5           displaying a graphical user interface (GUI) on a display;  
          receiving user input to the GUI indicating desired program operation of the graphical program;  
          executing a graphical program generation (GPG) program;  
          the GPG program receiving the user input, wherein the user input indicates  
10          desired operation of the graphical program;  
          the GPG program programmatically generating the graphical program in response  
to the user input indicating the desired operation of the graphical program, wherein the graphical program implements the desired operation.

15          19. The method of claim 18,

          wherein the GPG program comprises a graphical programming development environment application.

20          20. The method of claim 18,

          wherein the GPG program is operable to generate a plurality of graphical programs, depending on the received user input.

25          21. A computer-implemented method for programmatically generating a graphical program, the method comprising:

          displaying one or more input panels on a display;  
          receiving user input to the one or more input panels;  
          programmatically generating graphical source code for the graphical program,  
based on the received user input.

22. The method of claim 21,

wherein the one or more input panels comprise a graphical user interface (GUI) useable in guiding a user in specifying program functionality;

5 wherein the received user input specifies desired functionality of the graphical program;

wherein the programmatically generated graphical source code implements the specified desired functionality.

10

23. A computer-implemented method for programmatically generating a graphical program, the method comprising:

displaying a node in the graphical program in response to user input;

15 displaying a graphical user interface (GUI) for configuring functionality for the node in response to user input;

receiving user input via the GUI indicating desired functionality for the node;

programmatically including graphical source code associated with the node in the graphical program, wherein the programmatically included graphical source code implements the desired functionality.

20

24. The method of claim 23,

wherein said programmatically including graphical source code associated with the node in the graphical program comprises programmatically including the graphical source code as a sub-program of the graphical program, wherein the node represents the 25 sub-program.

25. The method of claim 23,

wherein said programmatically including graphical source code associated with the node in the graphical program comprises replacing the node in the graphical program with the programmatically included graphical source code.

5

26. A method for configuring a node in a graphical program, the method comprising:

displaying the node in the graphical program;

10 displaying a graphical user interface (GUI) associated with the node, wherein the GUI comprises information useable in guiding a user in specifying desired functionality for the node;

receiving user input to the GUI specifying desired functionality for the node;

15 programmatically generating graphical source code associated with the node to implement the specified functionality.

15

27. The method of claim 26,

wherein said programmatically generating graphical source code associated with the node comprises programmatically generating the graphical source code as a sub-program of the graphical program, wherein the node represents the sub-program.

20

28. The method of claim 26,

wherein said programmatically generating graphical source code associated with the node comprises replacing the node in the graphical program with the programmatically generated graphical source code.

25

29. The method of claim 26,

wherein no graphical source code is associated with the node until after said programmatically generating graphical source code associated with the node.

30. The method of claim 26,

wherein default graphical source code is associated with the node;

wherein said programmatically generating graphical source code associated with the node comprises replacing the default graphical source code with the programmatically generated graphical source code.

5

31. The method of claim 26,

wherein no functionality is defined for the node until after said programmatically generating graphical source code associated with the node.

10

32. The method of claim 26,

wherein no program instructions to be executed during execution of the graphical program are associated with the node until after said programmatically generating graphical source code associated with the node.

15

33. The method of claim 26, further comprising:

receiving user input requesting to change functionality of the node, after said programmatically generating the graphical source code;

re-displaying the GUI in response to the user input requesting to change functionality of the node;

receiving user input to the GUI specifying new functionality for the node;

programmatically replacing the previously generated graphical source code with new graphical source code to implement the new functionality for the node.

25

34. A memory medium for programmatically generating a graphical program, the memory medium comprising program instructions executable to:

display a graphical user interface (GUI) on a display;

receive user input to the GUI specifying desired functionality of the graphical program;

5 programmatically generate the graphical program in response to the user input specifying the functionality of the graphical program, wherein the graphical program implements the specified functionality.

35. The memory medium of claim 34,  
wherein the GUI comprises information useable in guiding a user in creation of a program.

10

36. The memory medium of claim 34,  
wherein the GUI comprises one or more GUI input panels;  
wherein the user input to the GUI comprises user input to each of the one or more GUI input panels.

15

37. The memory medium of claim 36, wherein the program instructions executable to display the GUI and the program instructions executable to receive user input to the GUI comprise program instructions executable to:

20 display a first GUI input panel on the display, wherein the first GUI input panel includes one or more first fields adapted to receive user input specifying first functionality of the graphical program;

25 receive first user input specifying first functionality of the graphical program;  
display a second GUI input panel on the display, wherein the second GUI input panel includes one or more second fields adapted to receive user input specifying second functionality of the graphical program;

receive second user input specifying second functionality of the graphical program.

38. The memory medium of claim 37,

wherein the second GUI input panel is one of a plurality of possible second GUI input panels, wherein the second GUI input panel is displayed based on the first user input.

5 39. The memory medium of claim 34,

wherein said programmatically generating the graphical program comprises programmatically generating a portion of a graphical program.

10 40. The memory medium of claim 34,

wherein said programmatically generating the graphical program creates the graphical program without any user input specifying the new graphical program during said creating.

15 41. The memory medium of claim 34,

wherein the programmatically generated graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program.

20 42. The memory medium of claim 34,

wherein the graphical program comprises a block diagram portion comprising a plurality of interconnected nodes and a user interface portion;

wherein said programmatically generating the graphical program includes generating the block diagram portion and the user interface portion.

25 43. The memory medium of claim 34, wherein the program instructions executable to programmatically generate the graphical program comprise program instructions executable to:

create a plurality of graphical program objects in the graphical program; and

interconnect the plurality of graphical program objects in the graphical program;

wherein the interconnected plurality of graphical program objects comprise at least a portion of the graphical program.

44. The memory medium of claim 34,  
5 wherein the user input received specifies an instrumentation function;  
wherein the programmatically generated graphical program implements the specified instrumentation function.

45. The memory medium of claim 44,  
10 wherein the instrumentation function comprises one or more of:  
a test and measurement function; or  
an industrial automation function.

15 46. A memory medium for configuring a node in a graphical program, the memory medium comprising program instructions executable to:  
display the node in the graphical program;  
display a graphical user interface (GUI) associated with the node, wherein the GUI comprises information useable in guiding a user in specifying desired functionality  
20 for the node;  
receive user input to the GUI specifying desired functionality for the node;  
programmatically generate graphical source code associated with the node to implement the specified functionality.

25 47. The memory medium of claim 46,  
wherein said programmatically generating graphical source code associated with the node comprises programmatically generating the graphical source code as a sub-program of the graphical program, wherein the node represents the sub-program.

48. The memory medium of claim 46,  
wherein said programmatically generating graphical source code associated with  
the node comprises replacing the node in the graphical program with the  
programmatically generated graphical source code.

5

49. The memory medium of claim 46,  
wherein no graphical source code is associated with the node until after said  
programmatically generating graphical source code associated with the node.

10

50. The memory medium of claim 46,  
wherein default graphical source code is associated with the node;  
wherein said programmatically generating graphical source code associated with  
the node comprises replacing the default graphical source code with the programmatically  
generated graphical source code.

15

51. The memory medium of claim 46, further comprising program instructions  
executable to:

receive user input requesting to change functionality of the node, after said  
programmatically generating the graphical source code;

20

re-display the GUI in response to the user input requesting to change functionality  
of the node;

receive user input to the GUI specifying new functionality for the node;  
programmatically replace the previously generated graphical source code with  
new graphical source code to implement the new functionality for the node.

25